

Mark scheme of Major 3 assessment

1	2	3	4	5	6	7	8	9
c	c	D	A	A	E	E	C	A

10.

- (a) The `Species` class inherits from the `Genus` class;
Accept "*Species extends Genus*".

[1 mark]

- (b) The `Specimen` class has a member variable that is a `Species` object;

[1 mark]

- (c) Award up to [4 marks max].
Correct visible 3 tiered diagram eg box diagram;
Correct class name in top box; Accept variations eg. "*Species extends Genus*".
Correct member variable in middle box;
ALL Correct member functions in lower box;

Species
String: speciesName
setSpeciesName(String s) String getSpeciesName() String toString() boolean equals(Species s)

[4 marks]

- (d) Award [1 mark] for identifying a benefit and [1 mark] for an elaboration, for two benefits, up to [4 marks max]

Benefits:

- Avoids duplicate code;
- Simplifies testing;
- Faster development;
- Can mimic real-world object relationships;

Accept other reasonable benefits

Example:

- Avoids duplicate code;
- Since all functionality of the parent class is inherited, it is not necessary to duplicate the code to reproduce that functionality in the new class;
- Simplifies testing;
- Functionality inherited from a parent class does not need to be re-tested in the new class;

[4 marks]

- (e) (i) The two methods are related to the specific class declared;
The compiler selects the correct method;

[2 marks]

- (ii) Polymorphism **OR** overriding;
Award [1 mark] for either.

[1 mark]

11.

- (a) Encapsulation refers to the practice of hiding the structure and representation of data within a class and making it accessible outside that class via accessor functions;

Accept "Refers to bundling the data and methods that operate on the data into one unit the methods and data can be accessed via dot notation or accessor methods."

[1 mark]

- (b) Award **[1 mark]** for identifying a benefit and **[1 mark]** for an elaboration, for two benefits, up to **[4 marks max]**.

Advantages:

- Data hiding and security;
- Ease of maintenance;
- Ease of testing;
- Speed of development;

Accept other reasonable advantages.

Example:

Advantage: Ease of testing;

By putting all the structure of the data in a single class, other classes which make use of that data can be easily tested by simply providing them with data values even before the "real" data class is available for testing;

Advantage: Ease of maintenance;

If there are changes to the data, or how it is stored or accessed internally, only the one class needs to be changed to accommodate that. All the other classes will continue to use the public accessor methods and will not need to be changed;

[4 marks]

- (c) Any accessor method in the `Specimen` class;

Example: `getName()`, `getCage()`, `getTOA()` are accessor methods in the `Specimen` class.

[1 mark]

- (d) Any instance variable in the `Specimen` class;

Example: `cageNumber`, `name` and `TOA` are instance variables in the `Specimen` class.

[1 mark]

- (e) *Award up to [3 marks max].*
Award [1 mark] for a constructor method requiring a single string as a parameter;
Award [1 mark] for String getGenusName() and a set method for the genus name;
Award [1 mark] for String toString() method that returns a valid string;

Example:

```
public class Genus
{
    private String genusName;
    public Genus(String g)
    {
        setGenusName(g);
    }
    public void setGenusName( String g ) { genusName = g; }
    public String getGenusName() { return genusName; }
    public String toString()
    {
        return "Genus: " + genusName;
    }
}
```

[3 marks]

- (f) *Accept any reasonable advantage and disadvantage. Award [1 mark] for identifying the (dis)advantage and [1 mark] for an elaboration, for one advantage and one disadvantage, up to [4 marks max].*

Example:

Advantage: The Specimen objects would inherit all the attributes of the Species object;

This would allow code in the Specimen object to access species-related methods directly rather than having to invoke them on a Species object using dot-notation;

Disadvantage: Logically inconsistent, in that data in the Species class may not be consistent across the associated Specimens;

Potential data duplication / waste of memory space;

Example 2:

Advantage: The Specimen object, as a subclass, will inherit all methods and instance variables applying to the Species object;

This would take up less space and make access to all common methods between specimens easier;

Disadvantage: Not all methods and variables are consistent between all specimens;

This means that changes made to methods or variables in the Species class may not apply to all specimens;

[4 marks]